Int. J. Engng Ed. Vol. 16, No. 1, pp. 21–38, 2000
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Knowledge-Based Engineering (KBE) Design Methodology at the Undergraduate and Graduate Levels*

D. E. CALKINS
University of Washington, Mechanical Engineering Department, Box 3526000, Seattle, WA 98195, USA
NATHANIEL EGGING and CHRISTIAN SCHOLZ
Sandia National Laboratories, MS 9105 P.O. Box 969, Livermore, CA 94550, USA. E-mail: cscholz@sandia.gov

An emerging design technology known as knowledge-based engineering (KBE) is the next step beyond CAD for product representation. KBE allows a true generative virtual prototype to be developed that represents both the geometric and the non-geometric characteristics of a product. Both an undergraduate and graduate level design course based on this technology is described. A new version of the design process is presented for the development of a virtual prototype. Examples of products (systems) that were modeled include a hand held vacuum and a parametric human which are presented and described.

BACKGROUND

THE DEVELOPMENT of complex systems requires a sequence of engineering and management decisions which must satisfy many competing requirements. Design is recognized as the primary contributor to the final product form, cost, reliability and market acceptance. The high-level engineering design and analysis process (conceptual design phase) is particularly important since the majority of the life-cycle costs and overall quality of the system are determined during this phase. The major opportunities for cost savings occur in the earliest phases of a product design. Approximately seventy per cent of the life-cycle costs are frozen by the end of the conceptual design phase, Fig. 1. The key to shortening the design cycle is to shorten the conceptual design phase, which will also reduce the amount of engineering in the redesign stage.

The engineering trade-off process during conceptual design is undertaken using good estimations and informal heuristics. Current traditional CAD tool support is extremely limited for the conceptual design phase. There is need to rapidly conduct design analyses involving multiple disciplines communicating together (trading off such things as performance, cost, reliability, etc.). Finally, it is necessary to be able to manage a large amount of domain-specific knowledge. The solution is to commit more resources at the conceptual design stage to reduce the cycle time by eliminating redesign.

All of these factors argue for an integrated design tool and environment that can help make decisions early in the design synthesis (conceptual design) process. This integrated design tool will enable a diverse and multi-disciplinary team of engineers, designers and stylists to achieve consensus of design intent under complex design requirements and increased design constraints. The design tool should allow the design team to examine more configurations at greater levels of detail. The problem then is to develop an architecture for a design tool that meets all of these requirements.

SYSTEMS DESIGN

Design process

A process is defined as an ordered set of steps that are performed to accomplish a task, i.e. the design of a product. The steps in a process that are intended to define how each step is to be accomplished. Design methodologies accomplish that task. While many models have been proposed for the design process, the one used attempts to incorporate current technology and tools available for design, Fig. 2. The process shown consists of six steps starting with 'Problem Definition' and ending with 'Prototyping.'

Design process stages

The design of a product traditionally proceeds through a series of well defined stages or phases including:

- conceptual design (concept exploration and development);

---

* Accepted 10 October 1999.

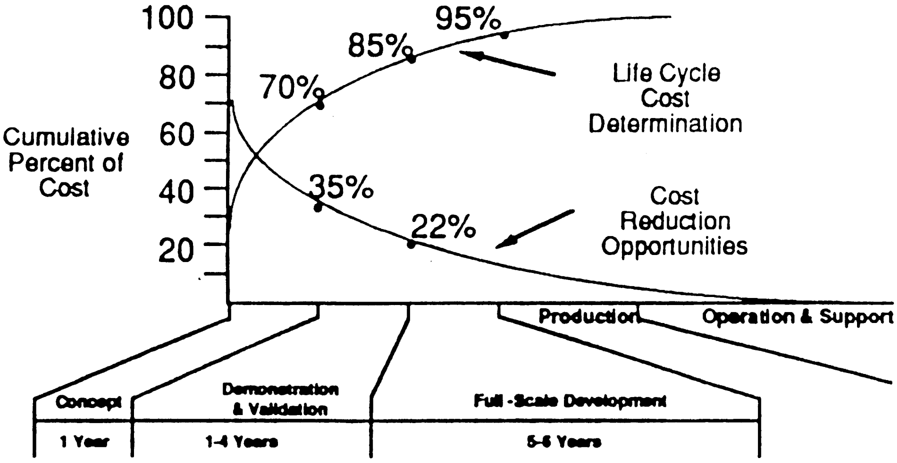


Fig. 1. Conceptual design effect on life cycle cost.

- preliminary design;
- detail design (production design).

The distinction between these stages is related to the level of design detail that is examined with regard to the system and its components.

Concept, or conceptual design, deals with development of a system at its very highest level, usually with a very coarse representation with only the major subsystems represented.

Preliminary design proceeds to the next level of representation and is also known as embodiment design.

Detail design includes analysis and results in a design description at a level suitable for manufacture. The arrangement, form, dimensions, tolerance and surface properties of individual parts are specified. Materials and manufacturing processes, and part assembly procedures, are also specified. Key factors in detail design are:

- standards
- standard components
- tolerances
- materials
- manufacturing processes.

Design types

Types of design include:

- parametric design
- routine design
- selection or component design
- prototype-oriented design.

Routine design, which comprises 80% of the engineering activity, is based on minor variations

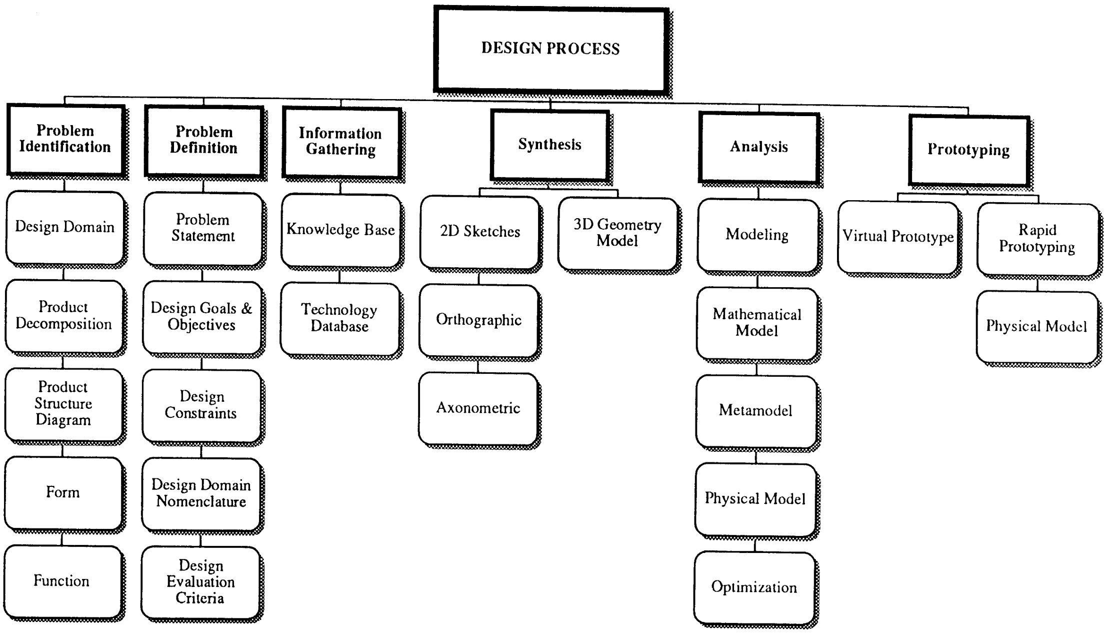


Fig. 2. Design process.

of pre-existing practice or procedure. **Knowledge-based engineering** (KBE) is the tool that is used for routine design where the product (system) is complex and has many assemblies, subassemblies, etc.

Current design tools

Many design tools have been developed to aid the design engineering in the process of design. These include 2-D CAD, 3-D CAD including wireframe, surface and solid modeling CAD, 2-D NURBS-based systems and parametric modeling, all of which are geometry based. CAD systems have been key in the development of drafting automation, but are not able to deal with knowledge such as rules, engineering practices or manufacturing processes. Parametric modeling CAD systems are based on the geometrical relationships between parts. Parts can be transformed by varying dimensions, thereby making changes easier and faster in the traditional design process. However, they do not easily manage the non-geometrical knowledge about a product design.

Product design knowledge must also include non-geometrical knowledge about the system performance to truly reflect the design intent. This aspect of the design process analysis has been traditionally addressed by applying procedural languages such as FORTRAN, Pascal or C. The procedural languages are powerful tools for engineering analysis applications, but have no inherent geometric capability. Another limitation of current CAD/CAE techniques is that they are not appropriate at the concept or early stage design level of complex systems.

VIRTUAL (DIGITAL) PROTOTYPE MODEL

What is needed is a way to represent the product design process to obtain a true virtual prototype which would allow the early development and evaluation of a product. The virtual prototype would replace traditional physical prototypes and allow the design engineer to examine 'what-if' scenarios while iteratively updating their designs. A true virtual prototype would not only represent the shape and form, i.e. the geometry, it would also represent non-geometric attributes such as weight, material, performance and manufacturing processes. Designers want a design representation that will be an exact representation of a physical prototype with both geometrical and non-geometrical attributes.

Product representation has moved from the 2-D orthographic drawing representation of the shape and form of the geometry, to full 3-D model representation of the geometry. The design tool that is needed for the design engineering domain, clearly must have attributes of all of the tools just discussed. It must combine the geometrical representation of the CAD systems, be able to do the engineering analysis of the procedural languages and represent the design knowledge as in an expert system. A true virtual prototype contains this full range of design knowledge.

ENABLING TECHNOLOGIES FOR DESIGN

Types of knowledge

'Knowledge', as applied to KBE, can be divided into four types [1]:

- facts,
- procedures,
- judgments,
- control.

Formalized knowledge found in handbooks such as material specifications, engineering data, ASTM standards, and equipment specifications is considered factual knowledge. Algorithmic and operative knowledge are the two forms of procedural knowledge. Numeric and non-numeric procedures for solving a problem or accomplishing some end are all elements of algorithmic procedural knowledge (APK). Facts are transformed by APK through engineering and analysis algorithms. Operative procedural knowledge (OPK) is used to create, delete, and transport facts. Examples of OPK programs are finite-element analysis, optimization, and database management systems [1]. Rules of thumb and common best-practices are examples of judgment knowledge. Heuristics, observations, experience, and plausible reasoning are also included in judgment knowledge. Logic and the formal principles of reasoning are fundamental to judgment knowledge application. Control knowledge is metaknowledge or knowledge about knowledge. The other types of knowledge are managed by control knowledge. Pattern directed actions, anticipation of unexpected developments, and dealing with uncertainties are all features of control knowledge [1].

Knowledge-based engineering (KBE)

The technology that allows the development of a true virtual prototype of a product is known as knowledge-based engineering, or KBE. KBE is the methodology for capturing and structuring knowledge about a design and its design process. KBE may be used to define engineering methods and procedures [2]. In KBE, the product structure tree (topology) is dynamic, so that KBE offers true engineering automation including application development, geometric modeling, application deployment and tools integration. Knowledge-based engineering is a programming tool used to develop a virtual prototype or a design advisor for the design of an established product in a given design domain. Dym, et al. [3] and Gonzalez, et al. [4] provide valuable overviews of KBE.

Existing knowledge about a class of designs is utilized in knowledge-based engineering or design (KBE or KBD) and organized into a database format usable by computers. Detailed designs or virtual prototypes are then rapidly developed through the use of digital computing power, developed databases, and systems of rules. The product model which is developed in the KBE environment is a **virtual prototype**. A virtual prototype has all of the geometric characteristics or attributes of the product as well as the non-geometric attributes such as materials, mass properties, stress and deflection characteristics, etc. Once the virtual prototype is created, it can be used by the designers to evaluate the success or merit of the design configuration, and then modify it if desired. The product model represents the engineering intent behind a geometric design. The information contained in a product model includes physical attributes like geometry, material type and functional constrains.

Generative technology

There are three types of KBE tools that are currently being explored and developed. These include:

1. Diagnostic approach (expert system).
2. Creative approach (design advisor)/(design checking).
3. Generative approach (virtual prototype).

The expert system was the first type of tool developed for use in the engineering domain. This tool is used for diagnostic purposes such as analyzing a malfunctioning automobile engine.

The second type, design advisor, is the one to more current developments. It is used to follow the design process of a system, and advise the designer of constraint and rules violations based on rules contained with the design advisor. The designer then acts on this advice and makes appropriate changes.

The third type involves developing a model of the system based on rules contained with the model. This model, a virtual prototype, then reacts to changes in attributes (either geometric or non-geometric), and regenerating a new instance of the prototype. This is the type of KBE that is used in the classes developed.

KBE uses generative technology to capture generic product design information, including geometry and topology, product structure development and manufacturing processes as design rules. Generative modeling maps functional specifications to a detailed representation of the product. The advantage of a generative model is that as the product requirements change, the design representation is immediately updated directly affecting all outputs. Thus, KBE is a dynamic object model wherein the representation of the design is continually updated. KBE methodology facilities the capture of engineering and manufacturing knowledge into a generative model by rapidly generating new designs from functional specifications. In contrast to the conventional design tools, KBE offers true design automation vs. design assistance. KBE is a robust design technology for continuous redesign of a system during the design process.

KBE product representation

Current KBE software is based on an object-oriented non-procedural design language such as LISP. As a result, the design information need not be ordered correctly within the model, as it will work out the order itself. Object-oriented programming works on the concept of objects that are used to represent the characteristics, both geometric and non-geometric, of actual physical objects. Objects are not passive, but can react with other objects. An object can create and store information and act in response to external stimuli. An object can demand information from another object, or send information to another object.

KBE enables true concurrent engineering by capturing the domain expertise of a range of contributors in an organization. This can include representatives from design, engineering, tooling and other areas of manufacturing. KBE vendors have a well-established methodology for capturing and codifying this range of product information. Often, KBE developers will collaborate with methodology consultants to learn the ‘knowledge capture’ process on a first development project and then will transfer and apply those skills to follow-on projects.

KBE tools

There are a variety of software tools available for KBE tool development. Included are ICAD^{TM}, TKSolver^{TM}, Design Link^{TM}, ProEngineer^{TM}, STONErule^{TM} and Smart Elements^{TM}. All of these are integrated with at least one of the contemporary CAD systems to provide a contemporary integrated design system. Unigraphics, CATIA^{TM}, Pro-Engineer^{TM}, IDEAS^{TM} and Auto-CAD^{TM} are some of the options.

These software tools are used to develop domain-specific design tools of the two KBE approaches, design advisor and the virtual prototype.

Generative virtual prototype (GVP)

The virtual prototype approach forms the basis of the KBE classes described, and is based on the use of the KBE software ICAD [5]. ICAD is used for meta-design, which is the design of design tools in the form of a product model. The product model is the framework for the product structure, engineering analysis, product cost, design standards, regulatory codes, material characteristics, manufacturing constrains and process plans. It is able to output a design report that represents the design state of the product. This report can include for example: data for analysis, 3-D geometric models, bills of material, cost reports and manufacturing instructions. The GVP captures and automates the functional design rules and understood methodologies of the engineering process. The GVP provides functionally valid alternatives for engineers to select and manipulate. The engineers add their judgement to optimize final systems designs.

A generative virtual prototype (GVP) is a system model that represents both the geometric and non-geometric attributes of a product (an object) which are embedded in the KBE model. It stores knowledge about a system in a product model composed of design and manufacturing engineering rules, which address both geometric and non-geometric issues. A generative virtual prototype is a combination of these design rules and includes a set of engineering instructions used to create the design, that is, the vehicle geometry. The generative virtual prototype represents the engineering intent behind the geometric design. It can store product information such as geometry and material specifications as well as process and performance information.

The generative virtual prototype paradigm is defined as follows:

Generative: generate or automatically produce an instance of the virtual prototype in response to an input state vector. Take input specifications, apply relevant procedures and generate a design automatically. When the requirements change, the design is updated immediately along with all of performance outputs.
Virtual: in effect although not in actual fact: a computer-based model
Prototype: original model or example of a particular type.

Design rules

KBE is based on the use of design knowledge in the form of 'design rules'. The design rules form the kernel of an object. Design rules comprise four basic categories:

1. *Heuristics*: comprised of experimental rules of thumb and 'best practices.' Usually based on corporate culture design heuristics. These are of the type, *If* (condition is true), *then* (action recommended).
2. *Empirical design rules*: these rules are based on curve-fitted expressions that are developed from experimental data. Meta-model technology used to develop models of complex systems.
3. *Legislated constraints*: these are comprised of rules established by law or by engineering standards.
4. *Laws of physics*: based on first principles in the form of analytical or numerical models. Also known as parametric rules. These rules are usually simple algorithms that would be solved using spreadsheet models.

Design rules are used to synthesize the knowledge in the knowledge base and to establish how the knowledge is used in a given model. The design rules are used to both define and relate the attributes in a KBE model. The methods and processes of an engineer are mimicked by these rules. Design rule types include:

- calculations
- conditionals
- look-up databases
- fixed
- variable
- references
- execute external programs
- selections
- optimizations.

KNOWLEDGE-BASED ENGINEERING UNDERGRADUATE AND GRADUATE DESIGN COURSES

KBE undergraduate and graduate courses

The Department of Mechanical Engineering at the University of Washington has recognized the importance of KBE in engineering, and both an undergraduate and a graduate level design class have been developed and offered. These include a special version of ME 495 senior level undergraduate capstone design class and ME 570, a graduate level design class. Both classes cover both KBE technology and its application by having the students go through the process of developing a product model, or virtual prototype.

Two different software tools were used in each class. The graduate course, which was developed and offered first, used the ICAD software with the support of Knowledge Technologies Inc. through the donation of the software ICAD. While ICAD is the industry standard tool for generative technology development, it has problems when used in the academic environment. It is expensive, and based on the UNIX environment, thus requiring an expensive workstation. We have only one system available for the class. A PC/NT-based solution was developed and tried in the undergraduate offering. This included the use of three software programs, TKSolver^{TM}/Design Link^{TM} for the rule base and ProEngineer^{TM} as the geometry engine. This proved to be a satisfactory low-cost solution for the single offering of the undergraduate class version.

Course description, format and grading process

The basis of the course is for a team of students, at each level, to develop a KBE generative virtual prototype. The difference between the undergraduate and graduate levels is the complexity of the system to be modeled, and the number of design rules that are embedded in the model. The course follows a seminar format with assigned readings and discussions. The students work together in design teams on a quarter-long design project. The student's grade depends on participation in the assigned group project. There are various technical article readings that are assigned during the quarter, as well as technical information packages. The student is expected to do research both at the engineering library as well, as on the Web. The students are formed into design teams of two and are expected to work together. At the end of the quarter, each team submits a formal design technical report and makes a formal technical presentation. Each team makes progress report presentations during the quarter and is graded by the other class

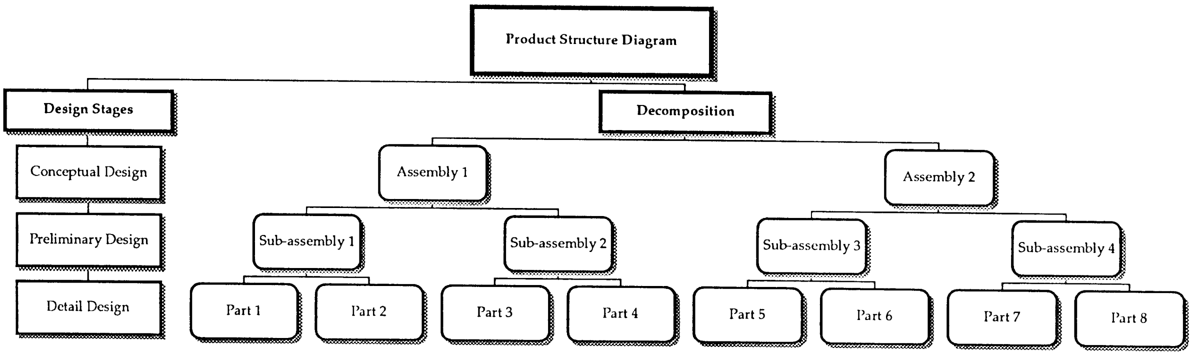


Fig. 3. Product structure diagram.

members on their oral presentation skills. The grade is based on technical content and communication skills (oral, written and graphical). The grade for this class depends on the following items:

- quality of work (technical content)
- writing communication skills
- oral presentation skills
- team participation

The students are informed that they can spend a great deal of time working long hours on the project. However their grade depends entirely on how they communicate that information and knowledge through their oral presentations and written reports. 'This is how it is in industry, and this is how it is in the class.' The students must also document their work thoroughly and keep a Design Notebook and Journal.

Course syllabus

1. Introduction to KBE
2. Design Process Models
3. Product Structure (Hierarchical) Decomposition
4. Design taxonomy: Environment, Problem & Process
5. Design Knowledge
6. Artificial Intelligence (AI)
7. Design Tools
8. Design Support System (DSS) Technology
9. Synthesis: Feature Based Parametric Geometry Modeling
10. Analysis: Simulation
11. KBE Tools

KBE COURSE TECHNOLOGIES

The course embodies several technologies in addition to its main focus of KBE. These technologies include product structure decomposition, metamodel technology and feature-based parametric geometry modeling.

Product structure decomposition

The concept of the virtual prototype may be related to the stages of the design process in its depth of representation. For example, if a product undergoes a process of decomposition down to its individual parts, Fig. 3, we see that each level of decomposition represents the product system at the various stages of design. For example, at its highest level, only the main sub-systems are represented and therefore correspond to the high-level or conceptual design stage. Thus the levels of detail of the design representation correspond to the development of the virtual prototype at the conceptual, preliminary and detail design stages.

This decomposition representation of the product is known as the product structure tree and represents the topology of the product.

Meta-model technology

Meta-model technology is used to develop the engineering analysis modules for the non-geometric attributes of the virtual prototype. Metamodels are approximating (empirical) models of engineering product or subsystem performance. They are used to approximate the response surface through the generation of design rules which are developed into meta-models to evaluate the performance of the main sub-systems.

Meta-models may be developed from heuristic knowledge using empirical data, or from more complex simulations such as FEA or CFD.

Feature-based parametric geometry modeler

Complex geometry may be modeled by decomposing the surface geometry into control curves which are defined by design 'features'. These 'features' include information on position, slope and curvature of each control curve. The value of these 'features' establishes a design state for the surface geometry. The control curve uses basis functions, usually polynomials, to define their shape. The feature value set is then used to quantify the control curves to generate the geometry. The designer must simply specify numerical values of the features to generate an instance of a system geometry.

KBE DESIGN PROCESS

A knowledge-based engineering design process was developed and used that reflects the architecture

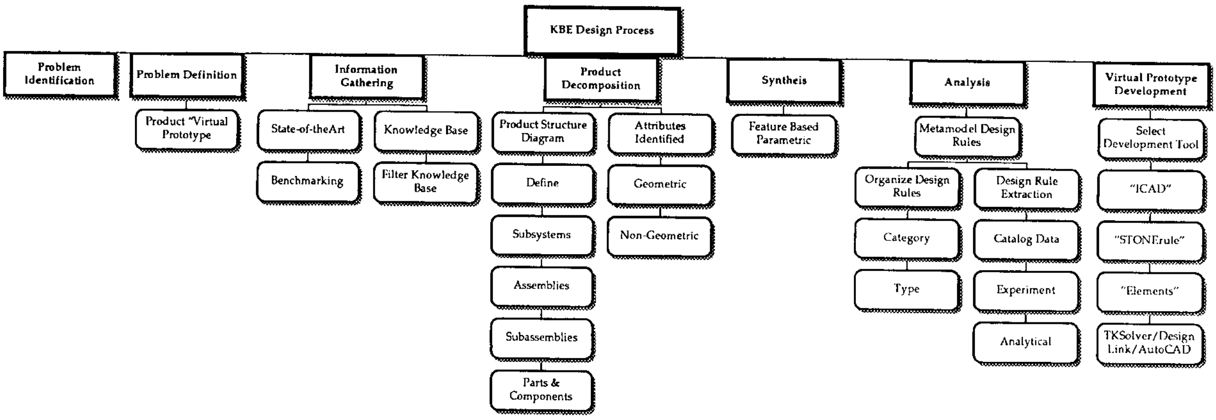


Fig. 4. Knowledge-based engineering (KBE) design process.

of the development of a KBE virtual product design process, Fig. 4. The process outlines the procedures that are used by the students over the quarter to guide them through the development of the virtual prototype.

Product decomposition

The product selected for the class virtual prototype exercise has to meet several criteria. First, a well established product was chosen so that multiple examples could be used for the product decomposition exercise. Second, the product must not be overly complex due to the time constraints of a ten-week quarter. Consumer products fulfilled these requirements. Thus far, two products have been used, a bicycle and a Black and Decker hand-vacuum.

Product decomposition is the process of disassembling a product and constructing a hierarchical representation of the product assemblies, subassemblies and parts. Product decomposition is a useful benchmarking tool as the assemblies and parts are analyzed and evaluated during the process. Therefore, decomposition is often used during the development of new or improved products to help engineers determine the state-of-the-art in a given design or application.

The purpose of decomposition is two-fold. First, the engineer is able to determine the form of the product and its component parts. The shapes of the physical entities that make up the product are described by the form. Second, insight is given by decomposition into the function of each entity. The tasks the entities perform is described by a function. The form and function of the Black and Decker VP300 hand-vac was determined by product decomposition. During decomposition, the assemblies, sub-assemblies, and parts were determined and the form and function of each were noted. The parts were then mapped out in a product decomposition chart, Fig. 5. The vacuum was made up of two assemblies, six subassemblies and thirty-four parts. A simple and efficient packaging design was exhibited by the vacuum. The materials used in the hand-vac, with the exception of the batteries and motor, were plastic, copper, bronze, and ceramic. Copper leads were used to conduct battery current. A 7.2 V Johnson motor was powered by two 3.6 V Nickel-Cadmium (NiCAD) rechargeable batteries. Suction was provided by a centrifugal fan.

Virtual prototype model scoping

Model scoping is the first step in the development of a virtual prototype KBE model. This is the process in which the range and scope of the KBE model are established by the engineer. As much or as little of the hierarchical tree as desired can be included in the KBE model. The entities within the chosen portion of the tree are incorporated into the virtual prototype. The level of the product structure diagram chosen to develop a VP corresponds closely to the stages in the design process. A top-level abstraction represents the product at the conceptual design stage, while a subassembly

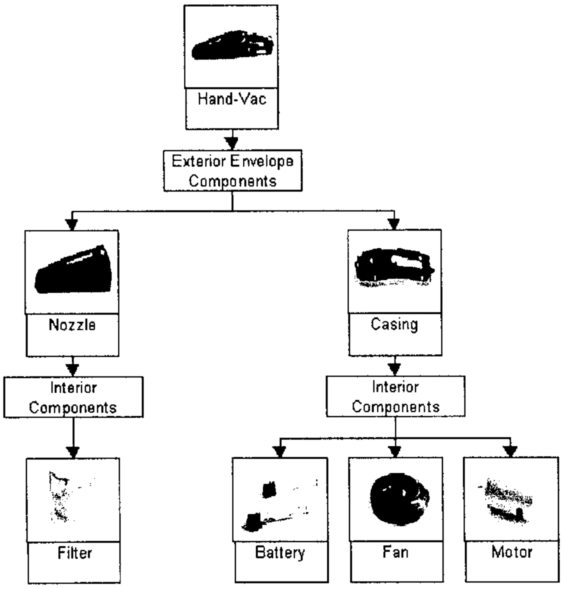


Fig. 5. Hand-vac virtual prototype conceptual design level decomposition.

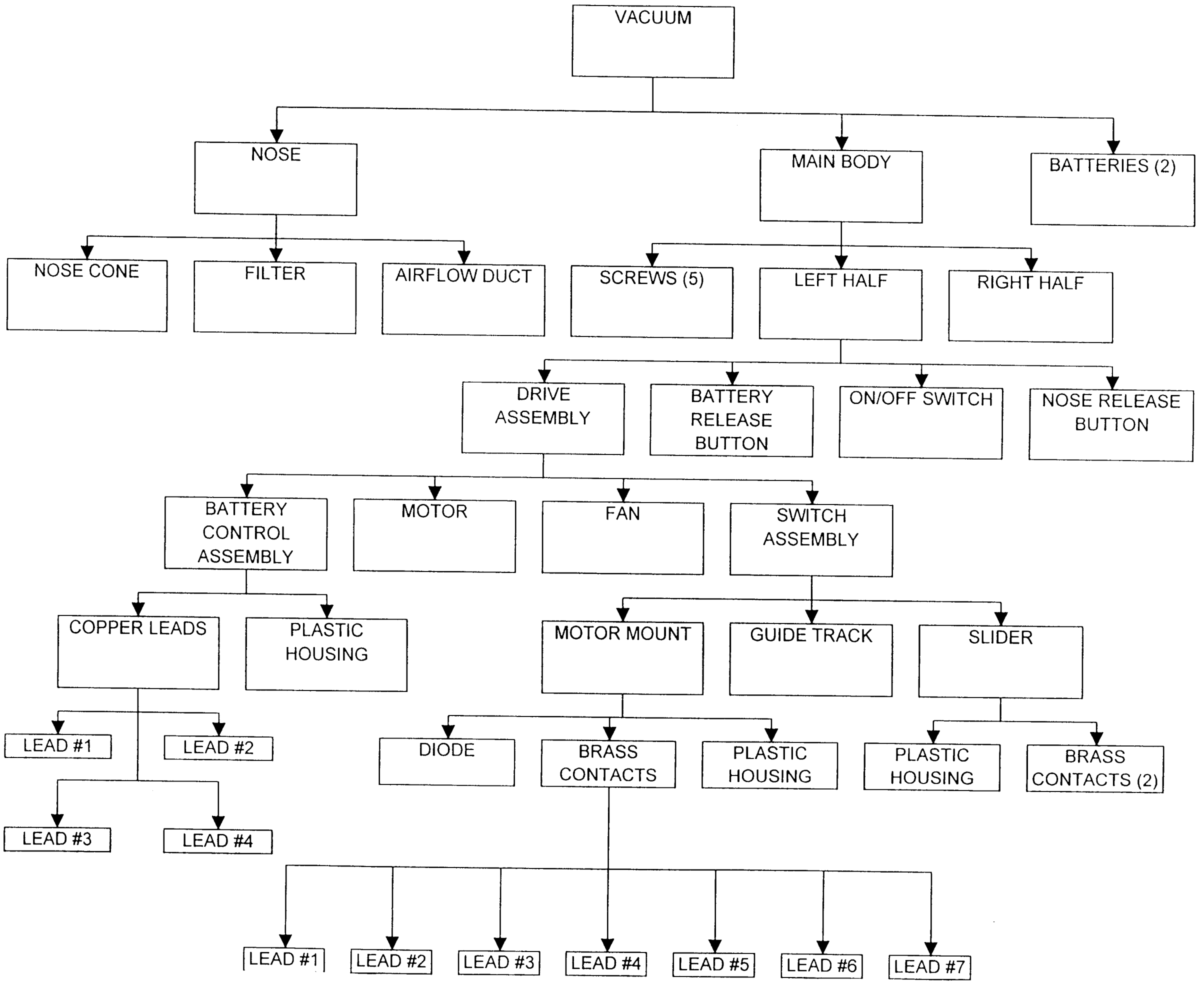


Fig. 6. Hand vac product decomposition (tree chart representation).

might represent the product at the detail design stage. In the case of the example used for the class, the top level was chosen, Fig. 6. The motor assembly, fan, batteries, case/handle, filter, and nozzle were included in the KBE model of the hand-vac, Fig. 7.

This was deemed a realistic model for the scope of the course which enabled the students to apply the principles of KBE within the confines of the academic quarter.

Geometrical and non-geometrical attributes

Geometrical and non-geometrical attributes are closely related to product form and function. The geometries of the product entities are described by the geometric attributes. Height, length, width,

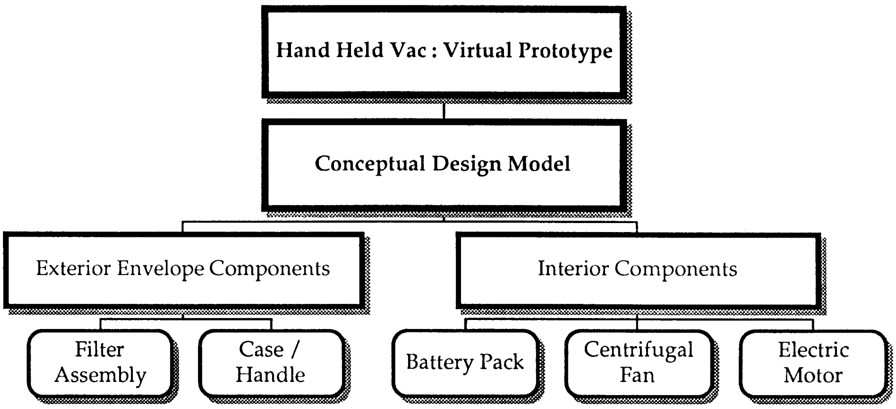


Fig. 7. Virtual prototype conceptual design level components.

volume, and area are some of these attributes. Attributes unrelated to the geometry are described by non-geometric attributes. Weight, material, function, cost, and manufacturability are included in the non-geometric attributes. The attributes are identified during product decomposition when the component parts and assemblies are most easily observed. Once the geometric and non-geometric attributes are identified, they are then examined to determine which attributes are interrelated. Readily changeable attributes are also identified.

An understanding of part attributes and their relationships is key to the development of a virtual prototype.

GENERATIVE VIRTUAL PROTOTYPE DEVELOPMENT PROCESS

Knowledge base development

Knowledge base development is the process of gathering knowledge and information pertinent to the design. The information necessary for the design and development of a given product virtual prototype is contained within the product knowledge base. Traditional and non-traditional means are techniques used to develop a knowledge base. Research, consulting, and benchmarking existing designs are included in traditional means. Solutions and company-specific solution techniques to past design problems are also considered a traditional knowledge-base development technique. Internet exploration and on-line data collection are considered non-traditional means.

Hand vacuum knowledge base

Due to limited time, a knowledge base was developed for only three of the six components chosen for the KBE model. These components involved the batteries, motor and fan. Battery research was conducted using the Internet, library and vendor data. The knowledge base was further added to by benchmarking the battery types used in a wide range of hand vacuums. A knowledge base for an extensive array of battery sizes, chemical compositions, and manufacturers was thus developed. Information regarding battery performance, capacity, size, cost and durability are contained within the knowledge base.

The knowledge bases for the motor and fan were developed in a similar manner. Information regarding motor sizes, specifications, and performance characteristics was obtained from the Internet. Information regarding fan performance and flow characteristics was obtained from the engineering library. In addition, tests were run on the fan/motor assembly to ascertain knowledge about the performance of the two components in conjunction. Torque, voltage, current, and velocity data were obtained in the tests. The test data were disseminated to the students via an information packet.

Knowledge base filtering

Knowledge base filtering is the process of filtering and condensing the knowledge base to extract the knowledge to be used directly in the KBE model. The knowledge base is filtered in a variety of ways. Trend analysis is used to filter large quantities of data by examining trends established among common data types. These trends can be observed by plotting the data; equations can be extracted from the plot and observed. Another filtering method is through selection. Selection is the process of selecting the knowledge necessary for a given KBE model. An example of selection is the narrowing of battery types to include only Nickel Metal Hydrides (NiMH) and NiCAD. In order for selection to take place, there must be some understanding of how the final model will work. As a greater understanding of the problem is gained, natural filtering occurs. As a result, knowledge that is unimportant or irrelevant to the model is abandoned.

The knowledge base for the hand vacuum was filtered using all of the above techniques. Data for the batteries was filtered using selection and trend analysis. NiMH and NiCAD batteries were chosen to be used in the model. Thus information regarding these batteries was selected from the knowledge base. NiMH and NiCAD battery sizes capable of powering a hand vacuum were also selected from the knowledge base. Trend analysis for the batteries was a two-phase process. Curve-fitting volume versus capacity for the selected battery types was accomplished in the first phase. A direct correlation between battery diameter and size (A, AA, AAA, etc.) was revealed in the second phase. Batteries of similar diameter were found to have similar overall characteristics. These batteries were grouped by size and averages were determined for capacity, mass, length, and mass density. The battery data were greatly reduced by this analysis.

The knowledge base for performance characteristics of the fan and motor were determined using another technique. In this case experimental tests were conducted to develop a database which in turn was used to formulate empirical relationships, or design rules. These empirical relationships form the basis of the fan/motor metamodel. These relationships included the motor/fan RPM as a function of input battery voltage and the current drawn as a function of the motor/fan rpm. It was shown that the performance characteristics of the fan and motor found in the vendor data were accurately represented by the fan/motor tests. Therefore, knowledge for the fan and motor was filtered to only include the relations developed during the fan/motor testing.

Design rule abstraction

Design rule abstraction is the process of abstracting or generalizing the rules and representation of parts within a particular model. All levels of modeling from individual parts to the entire assembly can be abstracted. A vacuum model containing only six components, motor, fan, handle/casing, filter, nozzle, and batteries, was developed through abstraction. These six components were all directly influenced by changes to the inputs of the KBE model. Those components associated with the chosen six that were not affected by the inputs were eliminated through abstraction. Wiring, switches, and screws were all eliminated in the abstraction process. The geometries of the six components were also abstracted. The batteries, motor, and fan were represented as cylinders. The casing, nozzle, and filter were represented as a box, truncated wedge, and truncated cone. In this manner, the model was simplified and made to focus on the important relationships within. There are many types of design rules within each rule category.

GRADUATE LEVEL HAND-VAC VIRTUAL PROTOTYPE DEVELOPMENT

Hand-vac parametric geometry model

Feature-based parametric modeling is a special class of design rules relating solely to geometry. The parameters are usually linear or radial dimensions, geometric relationships (tangent, concentric, parallel, etc.), or equations. In order to develop a true KBE model, a program must be used that allows both geometric and non-geometric features to be modeled; thus, a parametric CAD program is not suitable for KBE. As part of the development of the hand-vac model, geometric relationships were established linking the various parts and features of the hand-vac. First, the hand-vac was simplified by approximating it with various geometric primitives. These primitives were then mapped out parametrically and the relationships noted for use in the virtual prototype. The first letter of a part name is incorporated in its parametric labels. The casing, for example, has a hole in one side with a diameter labeled 'C_6'. In the parameter list C_6 is listed as equaling I_3, which is the exterior diameter of the impeller part. These relationships were incorporated into the virtual prototype to establish the geometric rules needed.

Hand-vac inputs and outputs

The determination of the inputs and outputs of a particular model is the first step in design rule development. The particular attributes or functional values are entered into the model as inputs to generate a certain output. All steps in design rule development are facilitated by the input/output determination. The goal is to correctly develop the design rules so that proper output is attained given the input to the model. The inputs for the hand-vac were divided into customer and manufacturer-driven categories. As the product being modeled is a home appliance, it was felt that the model input should reflect the concerns of customer focus groups that assist in the product design. Both manufacturer and customer concerns are addressed by establishing inputs for each. The following inputs were established for the hand vacuum:

Customer-driven inputs are:

- *Run time*: desired run time of the hand vacuum under full load, (min.)
- *Nozzle capacity*: dirt capacity the nozzle can hold, (in^3)
- *Suction pressure*: pressure at the nozzle tip, (psi)

Manufacturer-driven inputs are:

- *Battery type*: NiMH or NiCAD
- *Battery size*: AAA, AA, A, etc
- *Unified/scattered battery grouping*: batteries placed in single or multiple locations
- *Casing thickness*: material thickness of the plastic case, (in.)

Issues concerning the average customer are reflected in the customer-driven inputs. Issues that impact production cost, material cost, and design concerns are reflected in the manufacturer-driven inputs. The outputs listed below are generated from the above inputs:

- Number of batteries
- Position of batteries (in.)
- Nozzle volume, (in.^{3})
- Nozzle intake area (in.^{2})
- Overall vacuum dimensions, (in.)
- Center of gravity location, (in.)
- Weight, (lb.)
- Cost, ($)
- Mass moments of inertia, (in.-lb.-sec.^{2})

Hand-vac design rule abstraction

Two factors are critical to the development of the design rules. The first is an understanding of the attributes and attribute relations of each part. The second is complete knowledge of the inputs and outputs of the model. Design rules are developed once these factors are accounted for. Design rules are developed in two phases. The first phase is rule generation for the attributes of the assemblies, subassemblies and parts. The proper design rule type is dependent on the attribute being modeled. The attributes in the model are manipulated by rules that are generated in the second phase. Changes are imparted on the attributes of a model given changes to the inputs. The proper type of manipulation rule is dependent on the output desired from the model. The hand vacuum rules were developed using the above criteria. Design rules for the attributes were established and the manipulation rules were developed to achieve the desired output. The rules developed for each of the six parts of the hand-vac are discussed below. Many of the rules are taken from the parametric geometry representation of the hand-vac. The design rule types and examples for the hand-vac are given in Table 1.

Table 1. Types of design rules used

Type	*Example*
Calculation	Force = Mass × Acceleration
Conditional	Thickness = If (Pressure .30) 0.25 else 0.125
Look-up database	Battery-capacity = look up capacity of selected battery from external file
Fixed	Filter-volume = 8.5 in^3
Variable	Number-of-batteries = any multiple of 3
References	Length of side A = (half(Length of side B))
Execute external program	Thickness = FEA program result
Selection	Battery = battery that meets capacity, voltage and weight constraints

Hand-Vac ICAD constructs

A superset of LISP, ICAD Design Language (IDL) is the programming language used in ICAD to allow users to construct KBE models of parts or systems using geometric or non-geometric rules. ICAD modeling includes the following constructs:

- *Defpart*: a defpart ('definition of a part') is a description of a component of a product model, where the product model is a collection of defparts. a defpart includes the following:
- *Input-attributes*: describes the inputs that must be specified to create an instance of the defpart.
- *Attributes*: describes the engineering rules in the defpart.
- *Parts*: describes the components of the defpart, the product structure tree that is created when a design instance of the defpart is generated.
- *Defun*: a defun ('define function') is used to define a function used to perform a repetitive calculation. Defining a function is preferable to duplicating the expression every time it is needed. Furthermore, a defun can be revised in one location.

Defparts are created by an ICAD user to represent parts, assemblies or non-geometric entities. The defparts can be linked to each other through design rules to form a cohesive model of a product. It is also possible to create functions in ICAD that expand the already large library of built-in functions. This is accomplished with defuns; once defined, a defun can be called just like any normal function. The program is extensively powerful and extensive training is required to utilize its full potential.

Hand-vac choice attributes

Choice attributes were incorporated to allow the user to modify aspects of the model without reinstantiating it. These attributes allowed users to enter new values into attribute fields or select them from lists. For example:

1. A battery size (AA, AAA, C, etc.) and chemical composition (NiMH or NiCad) was chosen from a list of possible choices by the user. This was stored in the battery-size attribute. The user's choice was matched to the appropriate entry in an external data table and the capacities, costs, dimensions, and weights of that selection were retrieved. If more flexibility was desired in the battery selection, entire tables of manufacturer data can be used instead of the generic tables used in the model.
2. The ': run-time_min' was the run-time desired in the model. The run-time was used to calculate the number of battery banks needed for a particular size of battery. The total number of batteries was determined by the number of battery banks.
3. The ': intake-pressure_psi' was the desired pressure at the nozzle intake. This pressure was then used to calculate the intake area and the length of the noz-intake part in the nozzle.
4. The thickness for the casing and nozzle walls was determined by the ': casing-thickness_in' attribute. The overall mass and packaging concerns were affected by this attribute.
5. The length of the nozzle was derived from the ': nozzle-capacity_cu-in' attribute. The nozzle was lengthened until the desired volume was met. As mentioned in the discussion of the filter defpart, the desired volume was added to the filter volume to achieve the true nozzle volume.
6. The user was able to force ICAD to keep the batteries in one group through the use of the ': force-unified-batteries?' attribute. By default, the batteries were put in the handle first and then in the lower part of the case.

Hand-vac query attributes

Query attributes were used to read data from tables. Query attributes can be used to retrieve entire tables or selected records from a table. Information regarding batteries from the catalog 'batteries.table' was retrieved using the ': user-battery-selection' query-attribute.

A different table for each battery size or chemical composition could be included in future versions of the virtual prototype.

Hand-vac attributes

Nearly fifty attributes are incorporated in the virtual prototype. The geometric and non-geometric attributes of the Black and Decker vacuum were established during decomposition, and the interrelated and changeable attributes also identified, Fig. 8. Dimensions and other information (densities, costs, etc.) passed on to the defparts were represented by these attributes. Many of these attributes could have been included in the defparts they define, but it was decided to centralize as many attributes as possible. This was done to allow easier review of the outputs. A review of the categories of attributes follows.

Information on the battery, motor, and fan costs, as well as the cost-per-weight of the casing material, was provided by the costing attributes. This information was passed down to each part in

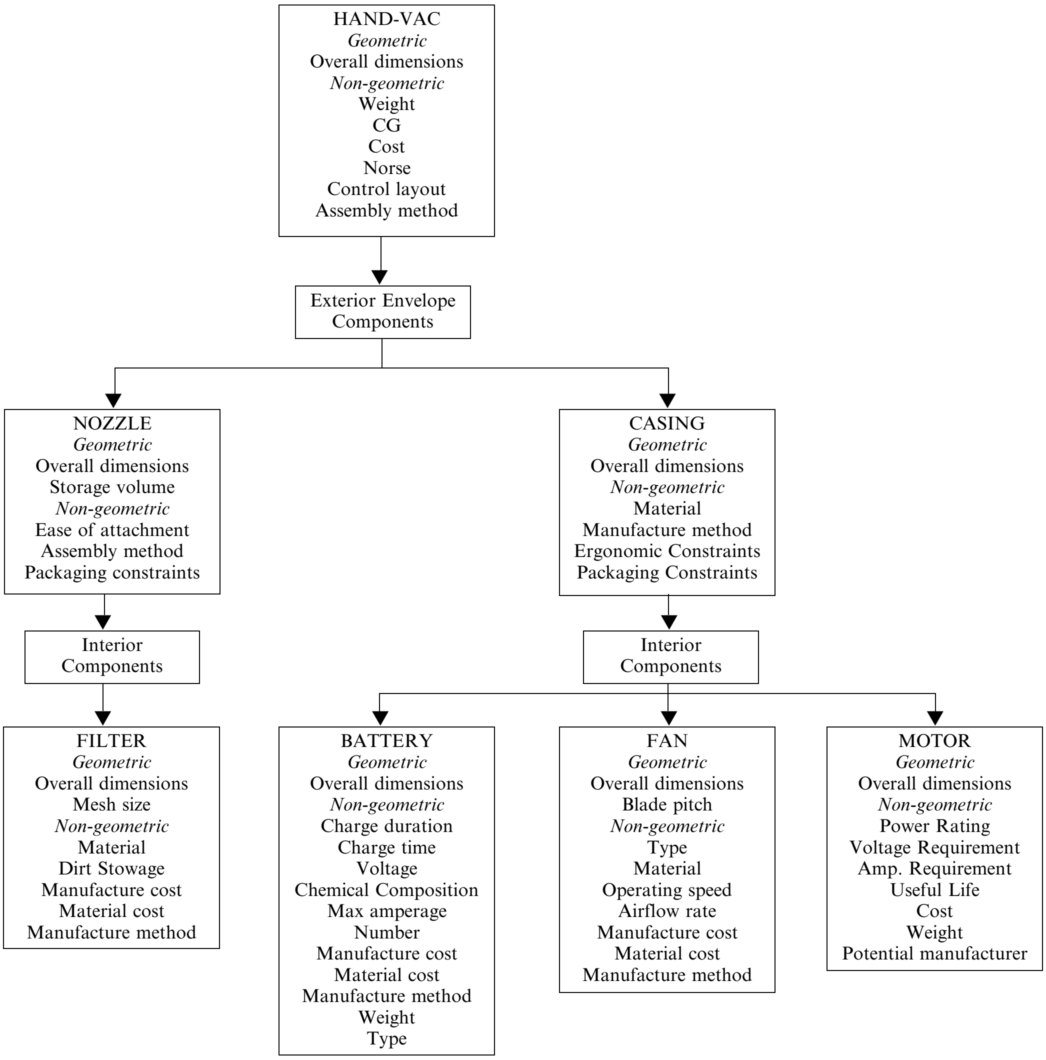


Fig. 8. Geometric and non-geometric attributes.

the tree and then summed up in the ': total-cost' attribute.

The densities for the various parts in Virtual Prototype were provided by the 'mass-property-info' attributes. These values were passed on to their respective parts. The center-of-gravity location for virtual prototype was found by the ': CoG' attribute.

The dimensions of the battery case part was determined by the battery casing attributes. If there were no batteries in the battery case, it was effectively removed. If batteries were present in the battery case, a decision was made by the code to determine whether one or two rows of batteries were needed. A battery case height sufficient to clear the batteries was also determined.

Battery data such as length, diameter, and density were retrieved from the query attributes by the battery specs attributes. The number of batteries and the number of battery banks were also determined by these attributes. The number of batteries increments in steps of 3, as three 1.2 V cells were required to run the 3.6 V motor.

The behavior of the motor and fan was modeled by the fan/motor attributes. Specifically, the operating current and intake pressure at maximum load were determined by these attributes. These values were used to calculate the number of banks needed in the battery specs attributes.

Placement of the batteries in the virtual prototype was achieved by the 'battery-placement-routine' attribute. The total number of batteries was split up into three separate values through a sequence of conditional statements. Each value was passed on to the various battery case/battery handle parts in the virtual prototype. First, an attempt was made by the virtual prototype to fit the batteries into the handle. If the handle length was exceeded by the total length of the batteries, the batteries were also placed in the lower battery case. If the 'unify batteries' command was given to Virtual Prototype, the batteries were only positioned in the batter case.

Lastly, a check by Virtual Prototype was run to determine whether the batteries could be positioned in an HCP arrangement within the battery

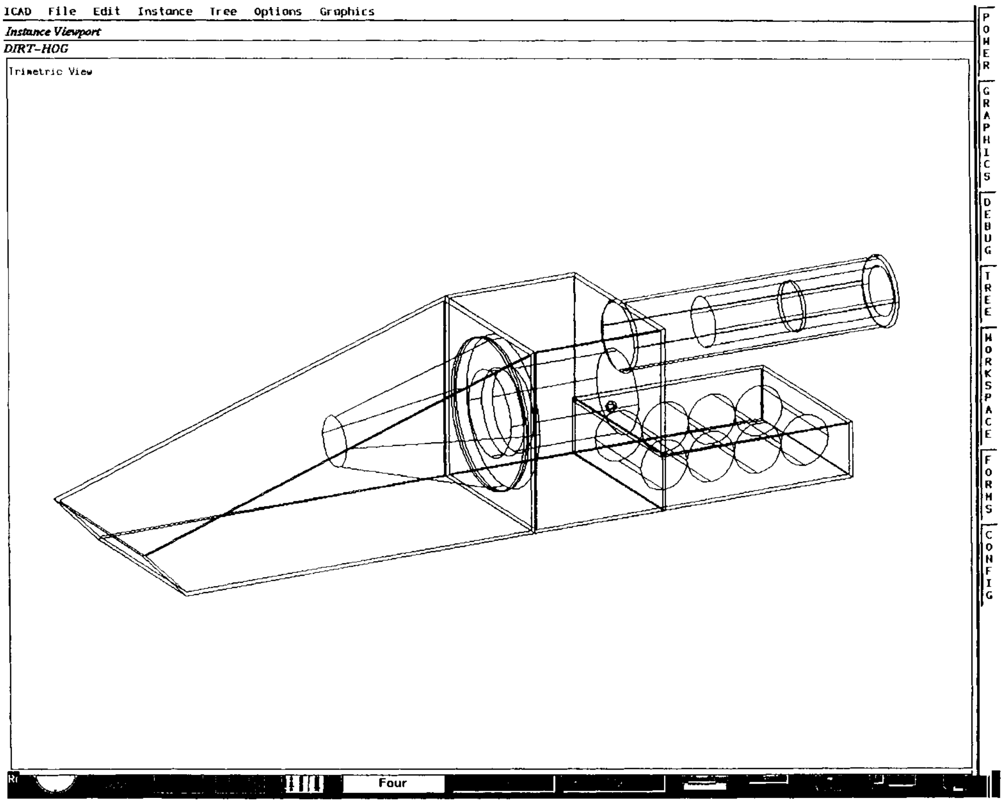


Fig. 9. Hand-vac case study #1.

case. This determination was dependent on whether the necessary handle and grip area was infringed upon by the battery case. The HCP arrangement was discarded if it resulted in too little room for a user to grasp the handle.

Hand-vac parts

Virtual Prototype was made up of nine parts; of these, eight were actual parts present in the hand-vacuum. The ninth part was a marker-sphere that indicates the center-of-gravity of the model. Null-parts were used to hide those parts that would disappear in a particular instance of the virtual prototype.

Hand-vac virtual design study

Three different hand-vac models were instantiated using the Dirt-Hog ICAD virtual prototype. Each model is described below. HandVac (1), Fig. 9, has been specified to use size 'C' NiMH batteries. The handvac features a non-unified battery packing arrangement, and due to the large size of the batteries, the four lower batteries are restricted to a single row instead of two stacked rows. The remaining two batteries have been placed in the handle, as a C size battery's diameter is small enough to fit. The nozzle has been specified to have a volume of 30 cubic inches. As can be seen, the Dirt Hog model has placed the batteries in appropriate locations and sizes the nozzle to provide the required capacity. The small sphere present in the approximate center of the vacuum is a marker indicating the center of gravity.

HandVac (2), Fig. 10, is identical to the one depicted in Fig. 9, except that the user forced the battery arrangement to be unified. This results in Dirt Hog placing all batteries in a single location. Very often, battery manufacturers will sell battery packs in this form and the model has the flexibility to reflect that possibility.

HandVac (3), Fig. 11, is also similar to the example in Fig. 9. It differs in that it uses size 'A' NiCad batteries. The lower energy density and reduced size result in a need for more batteries to achieve the same performance. However, due to the smaller size of the selected battery, Dirt Hog has decided to use an HCP arrangement for the lower seven batteries. This results in tighter packaging and a smaller vacuum than depicted in Figs 9 and 10.

UNDERGRADUATE LEVEL HAND VIRTUAL PROTOTYPE DEVELOPMENT

The focus on the undergraduate version of the KBE class was to develop and model a parametric human being. The intent was to actually develop a module of a KBE tool that will be used to design a complete automobile [6]. The students created both the human form geometry and the governing design rules. The human was modeled based on an

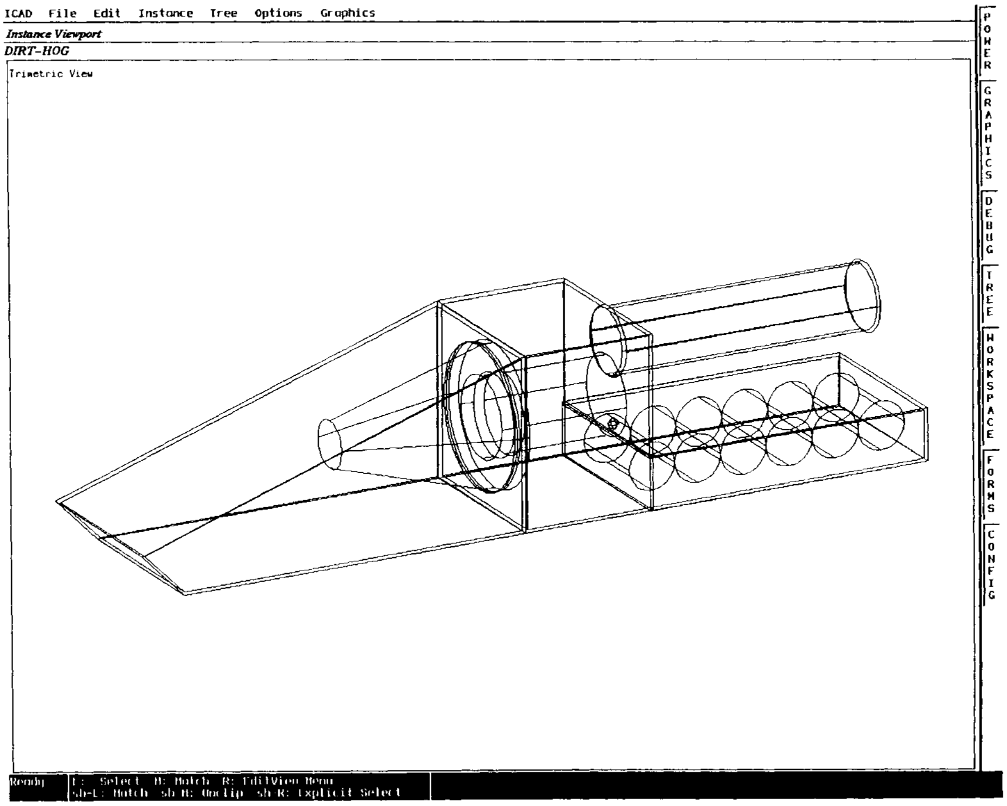


Fig. 10. Hand-vac case study #2.

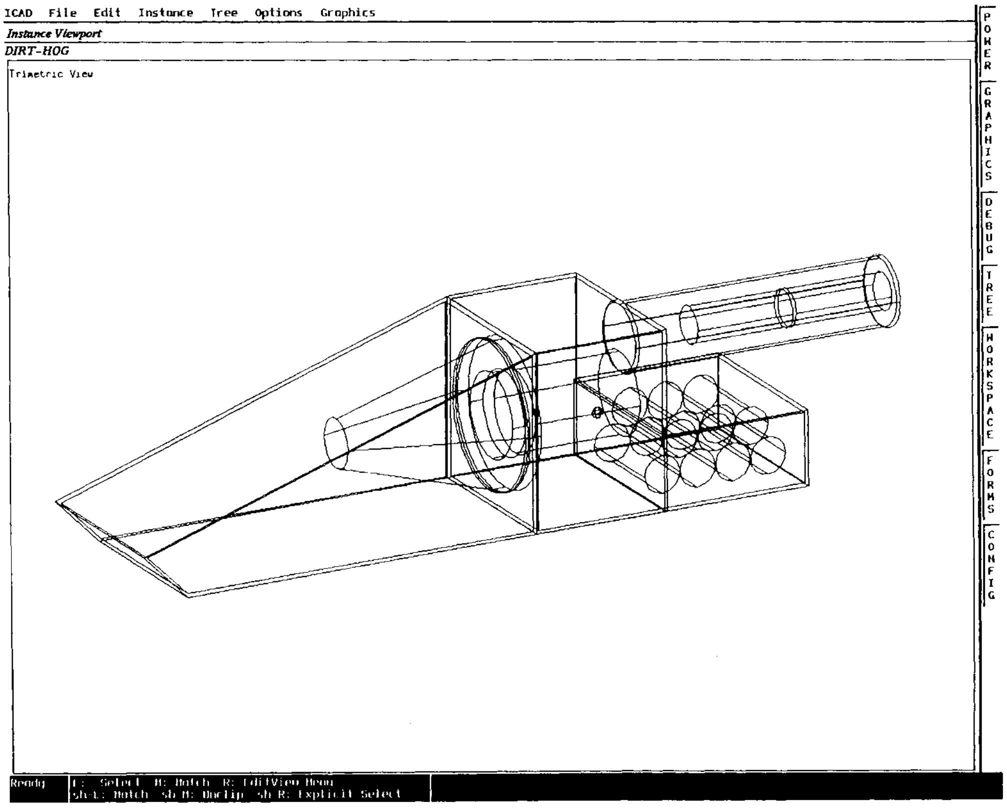


Fig. 11. Hand-vac case study #3.

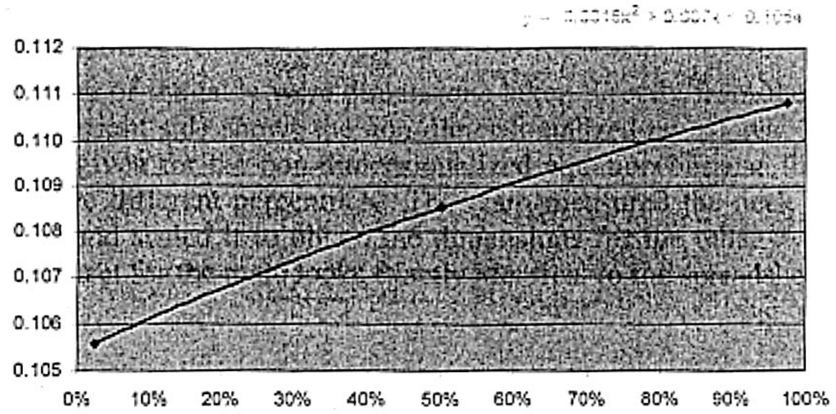


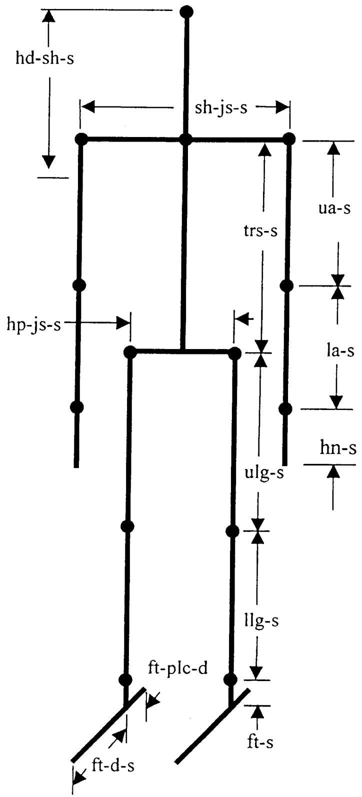
Fig. 12. Non-dimensional hand skeleton length.

ergonomic database developed by Dreyfus and Associates [7, 8] and Diffrient, et al. [9]. The database contains human geometry for both male and female gender and percentile (size). Percentiles included 2.5%, 50% and 97.5% for the entire dataset population.

The dimensions (lengths, widths and heights) were parameterized by dividing the dimensions from the database by the human height. The non-dimensional human dimensions were plotted against the percentile for both men and women and then curve-fitted to obtain the empirical design rules. Second-order polynomial fits were used for all design rules. All body parts were modeled in this fashion to obtain the complete skeleton model. For example the hand skeleton dimension, Fig. 12, is

$$Y = -0.0015 \times {}^{\wedge}2 + 0.007 + 0.1054 \qquad (1)$$

The basic human form was modeled using a combination of spheres, cylinders and extrusions the approximate to the overall anatomical proportions. Each feature presents a given body part. For example, the head was modeled using an ellipsoid and tapered cylinders were used to approximate the arms and legs with spheres at each joint. These basic forms were positioned on what was called the skeleton, Fig. 13. The skeleton was used to pose the human model, in a seated position for example.



Skeleton Definitions

Variable	Explanation
hd-sh-s	head-shoulder
sh-js-s	shoulder
ua-s	upper arm
la-s	lower arm
hn-s	hand
hp-js-s	hip-joint separation
trs-s	torso
ulg-s	upper leg
llg-s	lower leg
ft-s	foot
ft-d-s	foot depth
ft-plc-s	foot placement

Fig. 13. Skeleton structure.

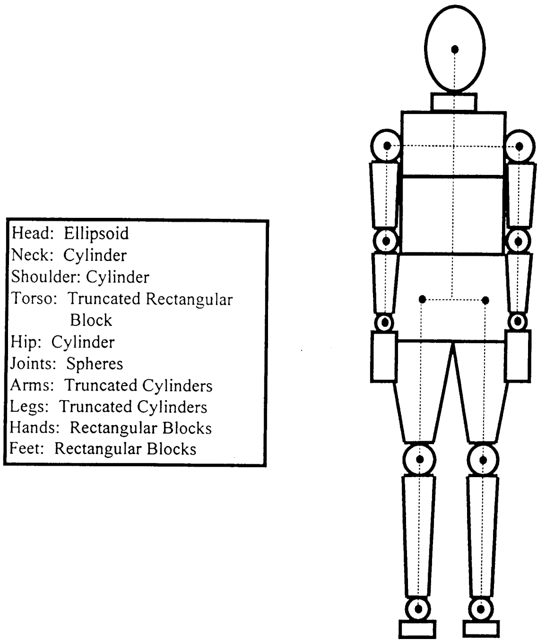


Fig. 14. Parametric human (skeleton with solid forms).

The 3-dimensional forms were then attached to the skeleton to form the complete human.

The skeleton geometry is shown in Fig. 14 along with the spherical joint location. The 3-dimensional forms are then added to form the complete human. The human model requires three inputs to create the overall human geometry in ProEngineer^{TM}. These include:

1. Gender (male/female).
2. Percentile, 2.5% to 97.5% (size).
3. Seat type (posture geometry) e.g. sport utility vehicle; sports car, coupe (two passenger); sedan (four passenger); coupe (four passenger).

The seat type specifies the seat posture geometry angles, Fig. 15. After specifying the desired human and seat in TKSolver, the human dimensions are calculated and sent to ProEngineer to generate the solid model representation, Fig. 16. An example instance of a 50th percentile male in a sports car is shown in Fig. 17.

CONCLUSIONS

The hand-vac virtual prototype was successfully developed by the graduate students. A knowledge base was developed for the motor, batteries, and fan. The hand-vac was then described through geometric and non-geometric attributes. Design rules were developed that incorporated these attributes and the knowledge base to allow construction of a virtual prototype. The prototype was programmed in the ICAD environment using the solid-modeler. The prototype was designed to allow the user to alter certain key attributes in the model to effect system-wide changes. Specifically, the user could alter battery size and chemical composition, battery packaging preferences, nozzle capacity, run time, and wall thickness of the casing. Based on these inputs, the prototype would alter performance characteristics, geometry and outputs such as weight and cost. This allowed the user to rapidly evaluate many different designs all sharing basic similarities.

The graduate class has completed its fourth offering with several new and exciting developments. The Boeing Co. actively supported the class with the award of two Boeing student internships for the summer 98 period. The class was divided into design teams of two students who worked on the class product design project. The final technical report and formal presentation at the end of the quarter were judged by Boeing engineers, and the winning team received the two internships. In addition, there were trips to the Boeing Co. to review their KBE projects, as well as in-class Boeing KBE specialists making presentations. In addition because of Boeing's interest in the class, the class has been opened up to graduate students in the departments of Civil and Aeronautical Engineering. Boeing has also requested that the class be opened up to senior undergraduate

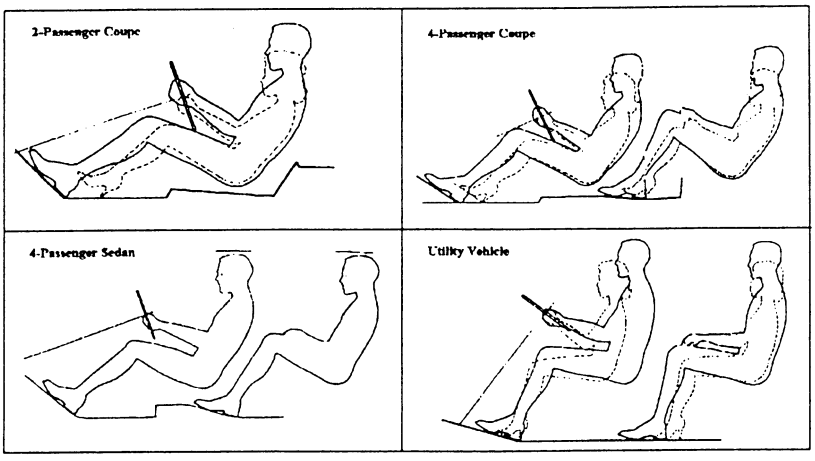


Fig. 15. Seating position geometry.

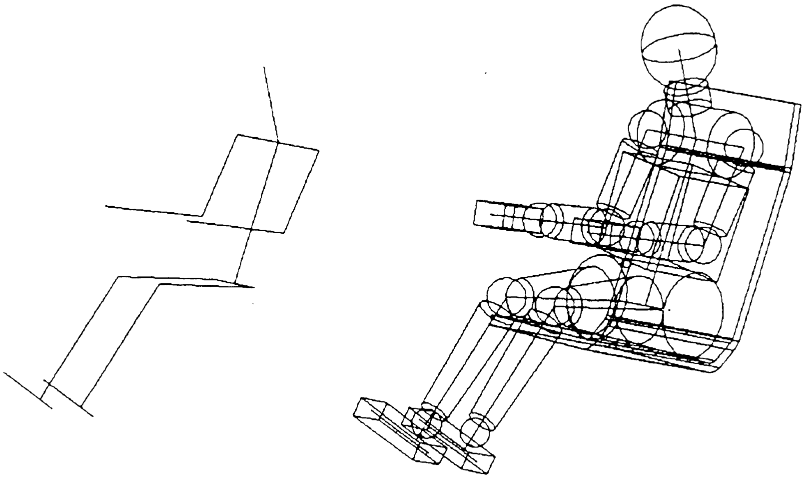


Fig. 16. Combined wireframe skeleton, solid forms and seat.

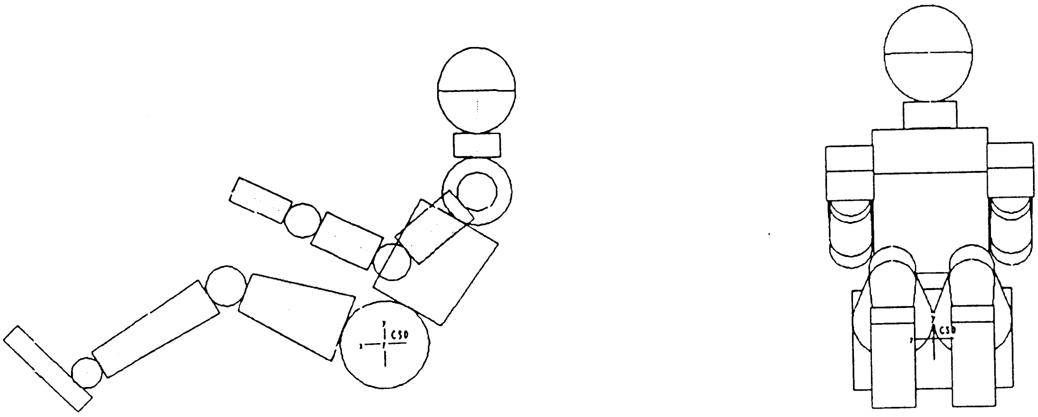


Fig. 17. 50th percentile male in sports car.

engineering students to increase the number of students exposed to the KBE technology. After learning about the Boeing internships, General Motors Truck and Pratt & Whitney each came up with two additional internships to be awarded to class members.

The parametric human virtual prototype was successfully developed by the graduate students. The PC/NT software suite chosen for the class was also successfully used. Because of the relative costs of the software and hardware. This approach will be adopted in future classes. In addition, the course will be offered to both undergraduate and graduate students simultaneously.

REFERENCES

1. W. J. Rasdorf, *Perspectives on Knowledge in Engineering Design*, Technical Report Article, Department of Civil Engineering and Computer Science, North Carolina State University, Raleigh North Carolina. pp. 249–253.
2. D. E. Calkins, Learning all about knowledge based engineering, in *Product Design and Development*, Chilton Co., (September 1996), pp. 30–31.
3. C. L. Dym and R. E. Levitt, *Knowledge Based Systems in Engineering*, McGraw-Hill, Inc. (1991).
4. A. J. Gonzalez and D. D. Dankel, *The Engineering of Knowledge-based Systems: Theory and Practice.* Prentice-Hill (1993).
5. M. R. Wagner, *Understanding the ICAD_System*, ICAD, Inc., Cambridge, MA (1999).
6. D. Calkins, W. Chan, and W. Su, A design rule based tool for automobile systems design, *SAE International Congress*, Detroit, MI (February 1998), TP 980397.
7. H. Dreyfus, *The Measure of Man: Human Factors in Design*, 2nd Edition, New York, Whitney Library of Design (1967).
8. Dreyfus and Associates, *The Measure of Man and Woman*, New York, Whitney Library of Design (1993).
9. N. Diffrient, A. Tilley and D. Harman, *Humanscale 7/8/9*, MIT Press, Cambridge, Mass. (1974, 1981 and 1991).

Dale E. Calkins was an Associate Professor of Mechanical Engineering at the University of Washington from 1979 until his death in 1999. He was a Lecturer at San Diego State University from 1977 to 1979 and a Visiting Professor at the Federal University of Rio de Janeiro, Brazil from 1976 to 1977. He worked as an engineer for Systems Exploration, Inc., San Diego, CA (1977–79), the US Naval Undersea Center, San Diego, CA (1964–73) and The Boeing Company, Seattle, WA (1961–64). His professional engineering career included academic, industrial, government and consulting experience in design engineering. His technical specialities included knowledge-based engineering, computer-aided design and engineering (CAD/CAE) and vehicle system design and analysis. He held the following degrees: D.Eng. in Naval Architecture from the University of California, Berkeley (1976); MS in Aerospace Engineering from San Diego State University (1969); and BS in Aeronautical Engineering from the University of Detroit (1961). He was a Registered Professional Engineer in the State of Washington. Professor Calkins passed away on Tuesday, June 29, 1999. He was 61 years old at the time of his death.

Christian Scholz is an engineer with Sandia National Laboratories, in Livermore, CA. He received his BSME in 1997 from the University of Oklahoma, in Norman, OK. His MSME research at the University of Washington focused on the development of a prototyp knowledge-based engineering automotive development program. During his graduate education, he was an instructor in the densior mechanical engineering capstone course, where he guided students in the design of a small Formula-style race cars for entry into the SAE Formula student design competition. He has participated in the development of five such vehicles, from 1995 through 1999.